APPENDIX

A. Metaevents

Sinfonia™ uses its own programming language to understand the way a song or show should behave. There are commands that tell Sinfonia™ how to navigate through a repeat or cut, which instruments to mute and when, which beat subdivision is to be tapped, etc. The language components are called metaevents, which serve to manipulate song files, without actually changing the underlying data.

While different metaevents involve different arguments and parameters, they share a common syntax. The first generation of Sinfonia™syntax elements are as follows:

1. *Measure*
   The measure where the action will occur.
2. *Beat*
   The beat where the action will occur.
3. *Tick*
   Each quarter note beat may consist of 480 ticks. The tick number specifies where within the beat the action will occur. For example, an action that happens on the second sixteenth note of a quarter note beat would be placed at tick 120.

The above three elements refer to location of the action. That is to say, the actions live within the chronology of the song or are expressly written to live outside the chronology (i.e. measure 0, beat 0, tick 0). Either way, all metaevents are placed somewhere on the song's timeline. For the purposes of expressing the other syntax elements, it will be assumed that metaevents are written at some location as expressed by measure, beat and tick.

1. Metaevent Structure

```
action type
wait
times
target
value
```

1. *Action Type*
   The basic command title (i.e. mute, repeat, cut). The individual actions will be explained in greater detail later in this section.
2. *Wait*
   If actions are placed within a repeated section of music, the wait field allows you to specify a number of times to wait before performing the action. For example, if you repeat measures 10-20 three times, you might tell Sinfonia™ to mute the flute the second time around by specifying a wait value of 1 (Sinfonia™ waits one time before performing the action). If the wait value is 0 or left blank, Sinfonia™ will assume you want the action to be performed every time.

3. *Times*

The times field tells Sinfonia™ how many times to perform an action. After Sinfonia™ has fulfilled its obligation and performed the action the specified number of times, it will ignore the metaevent on subsequent passes. The most common way to use this field is with repeat (i.e. repeat measures 10-20 three times). But there are other interesting ways to use it. For example, if measures 10-20 are repeated three times, and within this repeated section there are commands to mute the flute two times at measure 19 and unmute the flute at measure 20, Sinfonia™ would play the flute at measure 19 on the third pass after skipping it the first two times. If the times number is 0 or is left blank, Sinfonia™ will perform the action every time it is encountered.

4. *Target*

The target is the object of a given action. Targets vary according to action. For song navigation actions (cut, vamp, repeat, relocate, firstend, secondend) the target is the location where you will jump to when the action is performed. For show navigation actions (attacca, relseq), the target is the song to which you will relocate. For instrument actions (muteinstr, unmuteinstr, instrvolume) the target is the instrument you wish to manipulate. The targets are outlined with their corresponding action type below.

5. *Value*

Value is used with the instrvolume metaevent. This can be a number expressed relative to 1 with two decimal points. The value 1 indicates an unchanged volume level (or 100% of the preprogrammed value). Any value above or below 1 instructs Sinfonia™ to play a selected instrument relatively louder or softer. So for example, if an instrvolume action is added for the flute with a value of 1.25, Sinfonia™ will play the flute 25% louder than the preprogrammed level.

The following are first generation actions:

- Stop

When Sinfonia™ arrives at a *stop* action event, it will stop. Notes will cut off and the tempo clock will halt so that Sinfonia will not think you are switching to an extremely slow tap tempo (as it would if you simply stopped tapping). A *stop* metaevent is equivalent to hitting stop on the piano or typewriter keyboard. This might be useful if you want Sinfonia™ to stop at a specific spot for a bit of stage action.

Syntax: *stop* (wait) (times)

- Pause

Sinfonia™ will pause when it arrives at this action event. Forward motion of the song will stop, but whatever notes were playing when you arrived at the *pause* event will continue to play until you give Sinfonia™ further instructions (like additional taps or a cut-off command from the piano or typewriter keyboard). Like Stop, the tempo clock will freeze so that Sinfonia™ doesn't think you have shifted to a very slow tapping pattern. Pause events can be useful in musical fermatas or rubato passages.

Syntax: *pause* (wait) (times)

- Relocate

The *relocate* command is the generic way to move around a song. *Relocate* can accomplish the same tasks as any of the other song navigation commands (cut, repeat, vamp, etc.) The placement of the *relocate* action denotes the spot from which Sinfonia™ will jump to a new location. The target is the new measure to which Sinfonia™ will jump.

Syntax: *relocate* (target: location) (wait) (times)

- Vamp

Vamp events establish a new vamp, or section of music that is repeated an indefinite number of times. The vamped music will repeat until the player tells Sinfonia™ to exit the vamp. The *vamp* action is placed at the ending boundary of the vamped section of music and the target is the beginning. For example, if you wanted to add a new vamp of measures 1 through 4, you would place a *vamp* action at measure 5, beat 1, tick 0 (in other words – the place where measure 4 is completed, the barline to the right of m4). The target would then be 1 (for measure 1). The times value would be 0 or left blank since you would want the action to be performed indefinitely. (It is possible, though uncommon, that you might want a wait value – if for example this vamp fell within a larger repeated section of music).

Syntax: *vamp* (target: location) (wait)

- Repeat

Repeat events establish a new repeat. Like vamps, the repeat action occurs at the end point of the repeated section and the target is the beginning.

Syntax: *repeat* (target: location) (wait) (times)

- Firstend **and** Secondend

Target A identifies the beginning of the repeat section. This could also be a unique measure number. The metaevent structures *firstend* and *secondend* use the target in different ways. *Firstend* A will relocate to *secondend* A after the first time that it is approached. If the firstending is supposed to occur more than once, than the standard activation field can be used. *Secondend* A then is the point at which the sequence relocates back to Target A at the end of the firstending. The second time *secondend* is encountered it continues on.

- Relseq

The *relseq* action allows you to go immediately and automatically to a new song from a specific point in the current song. When the *relseq* is performed, Sinfonia™ will relocate to the selected song, stop and wait for further instructions (i.e. tap, go, etc.). Place the *relseq* action at the point you want to switch songs. The target will be the new song to which you wish to relocate. When creating or editing *relseq* actions, a drop down list of the various songs in the show will appear when you get to the target field. Click on your target choice.

Syntax: *relseq* (target: song) (wait) (times)

- Attacca

The *attacca* action allows you to segue between two songs without stopping. Usually this happens at the end of a given song such that the music flows from the end of one song into the beginning of the next song. However it's possible to place *attacca* anywhere in a song and to segue into any song in the show. Place the *attacca* action at the point you wish to segue with the destination song as your target. The available choices will appear in a drop down list when you get to the target field.

Syntax: *attacca* (target: song) (wait) (times)

- Reset

For actions that have either wait or times arguments, the actions will stop functioning or will function differently based on the number of times Sinfonia™ has passed. If you have a repeat to measure 10 that lives at measure 21 and it has a times of 2, than Sinfonia™ will relocate to measure 10 the first two times it hits measure 21 and ignore the command on the third and subsequent times. A *reset* action will reset all the action counters to their original values. So if Sinfonia™ encounters a *reset* action and then relocates to places where there are previously expired actions, Sinfonia™ will perform these actions as if they are being passed for the first time

Syntax: *reset* (wait) (times)

[For rehearsal purposes, leaving and returning to the current song with the arrow keys (in the Player window) will also serve to reset all the action events.]

- Muteinstr

A *muteinstr* action will mute a selected instrument.

Syntax: *muteinstr* (wait) (times)

- Unmuteinstr

An *unmuteinstr* action will unmute a previously muted instrument.

Syntax: *unmuteinstr* (wait) (times)

- Atempo

When Sinfonia™ encounters an *atempo* command, it will reset the tempo clock to the preprogrammed tempo. This can be useful if you want to perform an extreme ritardando section of music followed by a quick shift back to the original tempo. Without the atempo action, Sinfonia™ would take a few taps to realize your intentions and catch up. The atempo action allows you to give Sinfonia™ a heads up.

Syntax: *atempo* (wait) (times)

- Instrvolume

Instrvolume actions allow you to adjust the relative volumes of instruments over the course of a song. Place the *instrvolume* event at the desired location. A drop down list at the target field allows you to specify the instrument target. Here (at last), you get to use the value field. A decimal figure above 1 will make the instrument relatively louder. A decimal figure below 1 will make the instrument relatively softer. Thus a *instrvolume* with a flute target and a value of 1.20 will make the flute 20% louder, while a value of 0.80 will make the flute 20% softer.

Syntax: *instrvolume* (target: instrument) (value) (wait) (times)

- Tap

The *tap* metaevent allows you to change the tap beat subdivision. The target is the tap value choice. The options appear in a drop down list when you get to the target field.

Syntax: *tap* (target: subdivision) (wait) (times)

- Cutoff

When Sinfonia™ encounters a *cutoff* action, it will turn off all notes that are playing at the time.

Syntax: *cutoff* (wait) (times)

- Cut

The *cut* action is a relocate action, usually used to skip sections of music. Place the *cut* action at the desired jumping point with a target destination measure number.

Syntax: *cut* (target) (wait) (times)

- Clickon

The *clickon* action establishes a click track, which sends an audible click that can be heard by key music personnel (i.e. conductor, drummer). The click can be used as an audible guide to keep the ensemble in synch with Sinfonia™.

Syntax: *clickon* (wait) (times)

- Clickoff

Clickoff will stop a click track that may have previously been established.

Syntax: *clickoff* (wait) (times)